Nonlinear Optimization & (Some) Linear Algebra

Sven Leyffer

Mathematics and Computer Division
Argonne National Laboratory

CLAPDE LMS Symposium, July 19, 2008

Nonlinear Optimization & (Some) Linear Algebra

1. Introduction to Nonlinear Optimization

2. Local Methods for Nonlinear Optimization

   Active-Set Methods

   Interior-Point Methods

3. Forcing Strategy & Step Acceptance

   Penalty Functions

   Filter Methods for Optimization

   Funnel or Tolerance Tube

Introduction to Nonlinear Optimization

Nonlinear Programming (NLP) problem

$$(P) \begin{cases} \underset{x}{\text{minimize}} & f(x) & \text{objective} \\ \text{subject to} & c(x) \geq 0 & \text{constraints} \end{cases}$$

... $c(x) = 0$ are easy. Inequalities more powerful modeling paradigm.

- $f : R^n \to R$, $c : R^n \to R^m$ smooth (typically $\mathcal{C}^2$)
- $x \in R^n$ finite dimensional (may be large)
- more general $l \leq c(x) \leq u$ possible

Introduce slacks $s = c(x),\ s \geq 0$ and write as (re-define x)

$$(P)\ \underset{x}{\text{minimize}}\ f(x) \quad \text{subject to } c(x) = 0,\ x \geq 0.$$

Solving Nonlinear Optimization Problems

$$(P) \quad \underset{x}{\text{minimize}} \; f(x) \quad \text{subject to } c(x) \geq 0$$

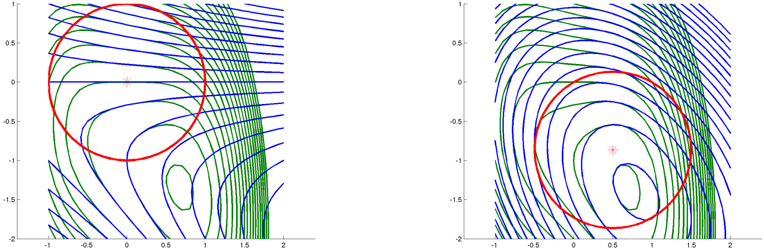
Main ingredients of iterative solution approaches:

1. Local Method: Given x_k (solution guess) find a step s.
   - Local problem should be easier to solve than (P).
   - Ensure fast (quadratic) local convergence.
   - Connection to global convergence ...
2. Forcing Strategy: Global convergence from remote starting points.
3. Forcing Mechanism: Truncate step s to force progress:
   - Trust-region to restrict s of local problem ... used in this talk.
   - Back-tracking line-search along step s.

Trust Region Methods

Unconstrained $f(x)$ minimization by trust-region

$$\underset{s}{\text{minimize}} \;\; q_k(s) := f(x_k) + \nabla f(x_k)^T s + \frac{1}{2} s^T H(x_k) s \text{ subject to } \|s\| \leq \Delta^k$$



Trust-Region Framework for Nonlinear Optimization

Nonlinear optimization problem

$$\underset{x}{\text{minimize}} \; f(x) \quad \text{subject to } c(x) \geq 0$$

Given x_0 starting point, set $k = 0$

REPEAT

1. solve trust-region problem around x_k for step s
2. **IF** $x_k + s$ improves on x_k **THEN**

   accept step: $x_{k+1} = x_k + s$

   else reject step: $x_{k+1} = x_k$
3. $k = k + 1$ & house-keeping

UNTIL convergence

Nonlinear Optimization & (Some) Linear Algebra

1. Introduction to Nonlinear Optimization

2. Local Methods for Nonlinear Optimization
   - Active-Set Methods
   - Interior-Point Methods

3. Forcing Strategy & Step Acceptance
   - Penalty Functions
   - Filter Methods for Optimization
   - Funnel or Tolerance Tube

Active-Set Methods

Sequential Quadratic Programming $\simeq$ Newton

Nonlinear optimization problem

$$\underset{x}{\text{minimize}} \; f(x) \quad \text{subject to } c(x) \geq 0$$

... linear model of constraints, quadratic model of objective:

$$\begin{cases} \underset{s}{\text{minimize}} & {g_k}^T s + \frac{1}{2} s^T H_k s \; := \; q_k(s) \\ \text{subject to} & c_k + {A_k}^T s \geq 0 \qquad y_k \geq 0 \\ & \|s\|_\infty \leq \Delta_k \qquad \text{trust-region} \end{cases}$$

where $y_k^T (c_k + {A_k}^T s) = 0$, complementarity.
Function gradient: $g_k = \nabla f(x_k)$,
Jacobian matrix: $A_k = \nabla c(x_k)^T$,
Hessian matrix: $H_k = \nabla^2 \mathcal{L}(x_k, y_k) = \nabla^2 f(x_k) - \sum [y_{k-1}]_i \nabla^2 c_i(x_k)$.

Main Computational Effort of SQP Method

At every iteration SQP solves QP by active-set:

$$\underset{s}{\text{minimize}} \; g^T s + \frac{1}{2} s^T H s \quad \text{subject to } A^T s \geq -c$$

Sequence of equality QPs $\simeq$ augmented systems (add/delete row/col)

1. determine new $\mathcal{A}$ estimate of QP active set
2. update sparse basis factors of $[A_{\mathcal{A}} : V]^{-1} = \begin{bmatrix} Y^T \\ Z^T \end{bmatrix}$
3. update factors of dense reduced Hessian: $Z^T H Z$ bottleneck!
   ... can replace dense reduced Hessian factors by CG.
4. perform two solves with $[A_{\mathcal{A}} : V]$ and one with $Z^T H Z$

Sequential Linear/Quadratic Programming

Nonlinear optimization problem

$$\underset{x}{\text{minimize}} \;\; f(x) \quad \text{subject to } c(x) \geq 0$$

"Decompose" QP step into two steps:

1. LP model ($\exists$ fast solvers) ... solve for s_{LP} step:

$$\underset{s}{\text{minimize}} \;\; g_k^T s \quad \text{subject to } c_k + A_k^T s \geq 0, \; \|s\|_\infty \leq \Delta_k$$

   ... to estimate active set $\mathcal{A} = \{i : c_i + a_i^T s_{LP} = 0\}$

2. equality constrained QP with $A_{:,\mathcal{A}}$ full rank

$$\text{(EQP)} \begin{bmatrix} H & -A_{:,\mathcal{A}} \\ A_{:,\mathcal{A}}^T & \end{bmatrix} \begin{pmatrix} s \\ y_{\mathcal{A}} \end{pmatrix} = \begin{pmatrix} -g \\ -c_{\mathcal{A}} \end{pmatrix}$$

   ... for fast local convergence (Newton) ... + inertia control?

Active-Set Identification by SLP

Polyhedral trust-region makes LP warm-starts inefficient

$$\begin{array}{lll} \underset{s}{\text{minimize}} & g^T s & \\ \text{subject to} & c + A^T s \geq 0 & \\ & \|s\|_\infty \leq \Delta_k & \text{trust-region} \end{array}$$

Practical Experience with SLIQUE

- Active constraints $c + A^T s \geq 0$ settle down
- Many changes trust-region bounds $\|s\|_\infty \leq \Delta_k$
  $\Rightarrow$ LP solvers slow, even near solution

Regularized LP Subproblems

A simple idea: penalize ℓ_2 trust-region $\Rightarrow$ lift into objective ...

$$\begin{array}{ll} \underset{s}{\text{minimize}} & g^T s + \pi \frac{1}{2} s^T s \\ \text{subject to} & c + A^T s \geq 0 \end{array}$$

Proximal point term $\pi \frac{1}{2} s^T s$

Regularized LP Subproblems

A simple idea: penalize ℓ_2 trust-region $\Rightarrow$ lift into objective ...

$$\begin{array}{ll} \underset{s}{\text{minimize}} & \mu g^T s + \frac{1}{2} s^T s \\ \text{subject to} & c + A^T s \geq 0 \end{array}$$

Proximal point term $\pi \frac{1}{2} s^T s$ becomes $\mu = \pi^{-1}$

Regularized LP Subproblems

A simple idea: penalize ℓ_2 trust-region $\Rightarrow$ lift into objective ...

$$\begin{array}{ll} \underset{s}{\text{minimize}} & \mu g^T s + \frac{1}{2} s^T s \\ \text{subject to} & c + A^T s \geq 0 \end{array}$$

Proximal point term $\pi \frac{1}{2} s^T s$ becomes $\mu = \pi^{-1}$

Dual of TR problem ... $\boxed{\nabla_s \mathcal{L} = \mu g + s - Ay = 0}$ eliminate s

$$\begin{array}{ll} \underset{y}{\text{minimize}} & \frac{1}{2} y^T A^T A y - (c - \mu A^T g)^T y + \frac{\mu^2}{2} g^T g \\ \text{subject to} & y \geq 0 \end{array}$$

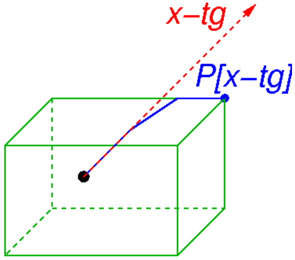
... bound constrained quadratic problem: $\exists$ matrix-free solvers ...

Projected Gradient CG for Bound Constraints

Dual of regularized LP:

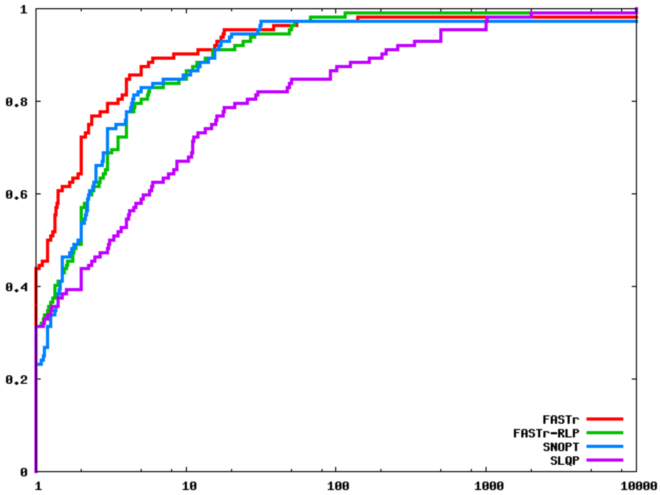
$$\underset{y}{\text{minimize}} \quad q(y) \quad \text{subject to } y \geq 0$$

... bound constrained quadratic
Projected gradient $P[x - \alpha \nabla q(y)]$
piecewise linear path
... large changes to $\mathcal{A}$-set
... but slow (steepest descent)



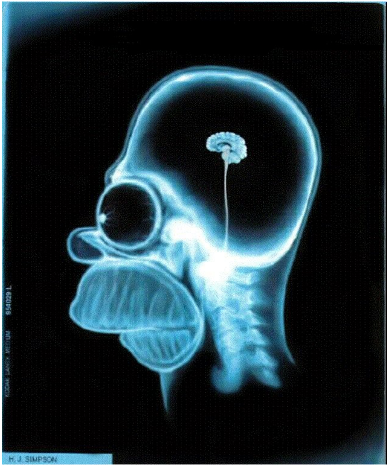
After each steepest descent step, minimize $q(y)$ on face $(\mathcal{A}_k)$
$\Rightarrow$ CG on inactive variables $(y_i > 0)$
... CG for $A^T A$, where A is sparse, may be rank-deficient.

Performance Profile: Active-Set Identification



... encouraging ...

Interior-Point Methods



Modern Interior Point Methods (IPM)

General NLP (with slacks)

$$\underset{x}{\text{minimize}} \;\; f(x) \quad \text{subject to} \quad c(x) = 0 \quad \& \quad x \geq 0$$

Perturbed $\mu > 0$ optimality conditions ($x, z > 0$)

$$F_\mu(x, y, z) = \left\{ \begin{array}{r} \nabla f(x) - \nabla c(x)^T y - z \\ c(x) \\ Xz - \mu e \end{array} \right\} = 0$$

- Primal-dual formulation, where $X = \text{diag}(x)$
- Central path $\{x(\mu), y(\mu), z(\mu) \;:\; \mu > 0\}$
- Apply Newton's method for sequence $\mu \searrow 0$

Modern Interior Point Methods (IPM)

Newton's method applied to primal-dual system ...

$$\begin{bmatrix} \nabla^2 \mathcal{L}_k & -A_k & -I \\ A_k^T & 0 & 0 \\ Z_k & 0 & X_k \end{bmatrix} \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \end{pmatrix} = -F_\mu(x_k, y_k, z_k)$$

where $A_k = \nabla c(x_k)^T$, X_k diagonal matrix of x_k.

- Polynomial run-time guarantee for convex problems
- Need $\mu \searrow 0$ to converge nonlinear optimization
  $\Rightarrow$ systems becomes more ill-conditioned $\mathcal{O}(\mu^{-1})$
  ... want higher accuracy for smaller μ
- Constraint preconditioners avoid ill-conditioning
  ... other techniques aim to identify active constraints

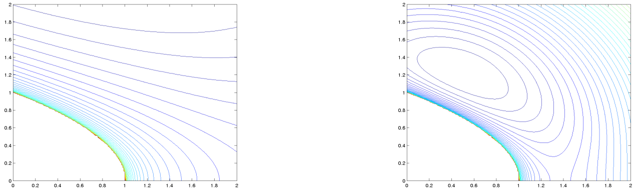
Classical Interior Point Methods (IPM)

$$\underset{x}{\text{minimize}}\ f(x) \quad \text{subject to } c(x) = 0 \quad \& \quad x \geq 0$$

Related to classical barrier methods [Fiacco & McCormick]

$$\begin{cases} \underset{x}{\text{minimize}} & f(x) - \mu \sum \log(x_i) \\ \text{subject to} & c(x) = 0 \end{cases}$$

$\mu = 10$ $\mu = 1$



$$\text{minimize } x_1^2 + x_2^2 \quad \text{subject to} \quad x_1 + x_2^2 \geq 1$$

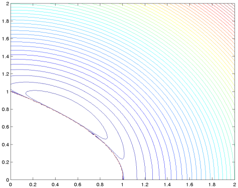
Classical Interior Point Methods (IPM)

$$\underset{x}{\text{minimize}} \; f(x) \quad \text{subject to } c(x) = 0 \quad \& \quad x \geq 0$$

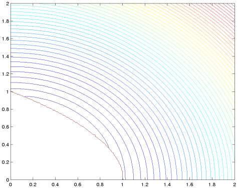
Related to classical barrier methods [Fiacco & McCormick]

$$\begin{cases} \underset{x}{\text{minimize}} & f(x) - \mu \sum \log(x_i) \\ \text{subject to} & c(x) = 0 \end{cases}$$

$\mu = 0.1$



$\mu = 0.001$



$$\text{minimize } x_1^2 + x_2^2 \quad \text{subject to} \quad x_1 + x_2^2 \geq 1$$

Nonlinear Optimization & (Some) Linear Algebra

1. Introduction to Nonlinear Optimization

2. Local Methods for Nonlinear Optimization
   - Active-Set Methods
   - Interior-Point Methods

3. Forcing Strategy & Step Acceptance
   - Penalty Functions
   - Filter Methods for Optimization
   - Funnel or Tolerance Tube

Enforcing Convergence

When's a New Point Better?

Easy for unconstrained minimize $f(x)$ (quadratic model $q_k(s)$):

$$x_{k+1} = x_k + s \ \text{ better, iff } \ f(x_{k+1}) \leq f(x_k) - 10^{-4} q_k(s)$$

... actual reduction matches portion of reduction predicted by model.

Unclear for constrained problem: $c^-(x) := \max(0, -c(x))$

- step s can reduce both $f(x)$ and $\|c^-(x)\|$ GOOD
- step s increases $f(x)$ and decreases $\|c^-(x)\|$???
- step s decreases $f(x)$ and increases $\|c^-(x)\|$???
- step s can increase both $f(x)$ and $\|c^-(x)\|$ BAD

Penalty Functions

$$(P) \qquad \underset{x}{\text{minimize}} \; f(x) \quad \text{subject to } c(x) \geq 0$$

Penalty function simplifies acceptance: $c^-(x) := \max(0, -c(x))$

$$(P_\pi) \qquad \underset{x}{\text{minimize}} \; \Phi(x, \pi) \; = \; f(x) + \pi \|c^-(x)\|_1$$

where $\pi > 0$ sufficiently large penalty parameter.

Theorem: If $\pi > \|y^*\|_\infty$ then $(P) \Leftrightarrow (P_\pi)$, where y^* optimal multipliers.

Classical penalty approach: $\pi_k = \|y_k\|_\infty + 1$ for $y_k \simeq y^*$ multipliers.

Penalty Functions & Methods

Modern penalty approach:

- Ensure π large enough to give descend of quadratic model:

$$q_\pi(s) = f_k + \nabla f_k^T s + \frac{1}{2} s^T H_k s + \pi \|(c_k + A_k^T s)^-\|_1$$

... require $q_\pi(0) - q_\pi(s) \geq \epsilon \left(\|c_k^-\|_1 - \|(c_k + A_k^T s)^-\|_1 \right)$

- Equivalent to ...

$$\pi_k \geq \frac{g_k^T s + \frac{\sigma}{2} s^T H_k s}{(1-\epsilon)\left(\|c_k^-\|_1 - \|(c_k + A_k^T s)^-\|_1 \right)}$$

where $\sigma = 1$ iff $s^T H_k s > 0$, and $\sigma = 0$ else.

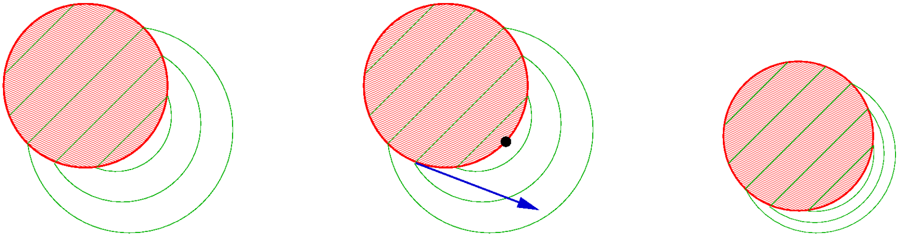
- Make sure that denominator $\neq 0$ ... threshold parameter.
- Works better than classical approach.

ℓ_1 Exact Penalty Function & Maratos Effect

$$\underset{x}{\text{minimize}} \;\; \Phi(x, \pi) \;=\; f(x) + \pi \| c^-(x) \|_1$$

where $c^-(x) := \max(0, -c(x))$ constraint violation

- Φ nonsmooth, but equivalent to smooth problem
- Penalty parameter not known a priori: $\pi > \|y^*\|_\infty$
- Large penalty parameter $\Rightarrow$ slow convergence; inefficient

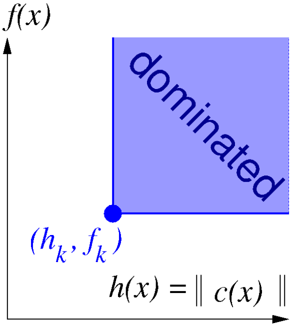


Maratos effect motivates second-order correction steps

Filter Methods to Promote Convergence

Penalty function combines two competing aims:

1. Minimize $f(x)$
2. Minimize $h(x) := \|c^-(x)\|$ ... more important



Borrow concept of domination from multi-objective optimization

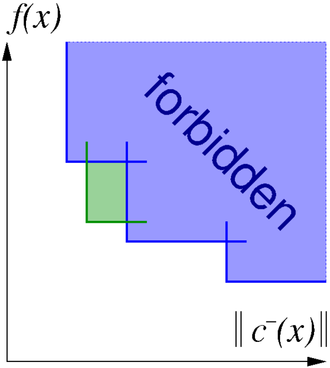
(h_k, f_k) dominates (h_l, f_l)
iff $\quad h_k \leq h_l$ & $f_k \leq f_l$

i.e. x_k at least as good as x_l

Filter Methods to Promote Convergence

Filter $\mathcal{F}$: list of non-dominated pairs (h_l, f_l)

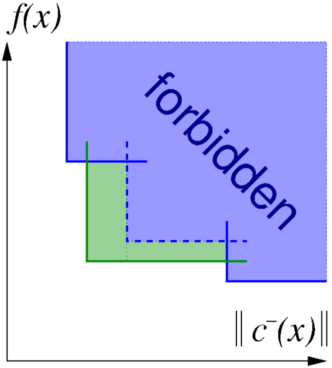
- new x_{k+1} acceptable to filter $\mathcal{F}$, iff
  1. $h_{k+1} \leq h_l \; \forall l \in \mathcal{F}$, or
  2. $f_{k+1} \leq f_l \; \forall l \in \mathcal{F}$



Filter Methods to Promote Convergence

Filter $\mathcal{F}$: list of non-dominated pairs (h_l, f_l)

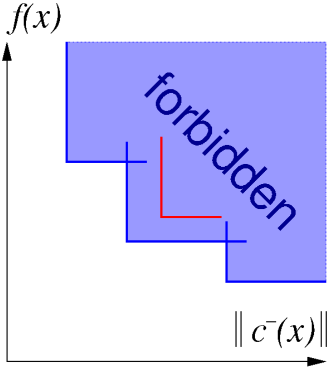
- new x_{k+1} acceptable to filter $\mathcal{F}$, iff
  1. $h_{k+1} \leq h_l \; \forall l \in \mathcal{F}$, or
  2. $f_{k+1} \leq f_l \; \forall l \in \mathcal{F}$
- remove redundant entries



Filter Methods to Promote Convergence

Filter $\mathcal{F}$: list of non-dominated pairs (h_l, f_l)

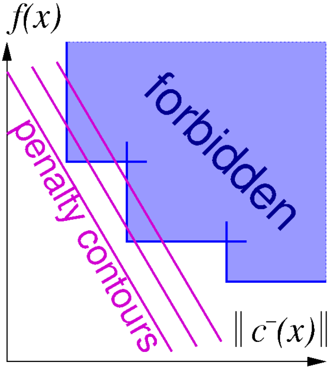
- new x_{k+1} acceptable to filter $\mathcal{F}$, iff
  1. $h_{k+1} \leq h_l \ \forall l \in \mathcal{F}$, or
  2. $f_{k+1} \leq f_l \ \forall l \in \mathcal{F}$
- remove redundant entries
- reject new x_{k+1}, if $h_{k+1} > h_l$ & $f_{k+1} > f_l$ & reduce trust region $\Delta = \Delta/2$



Filter Methods to Promote Convergence

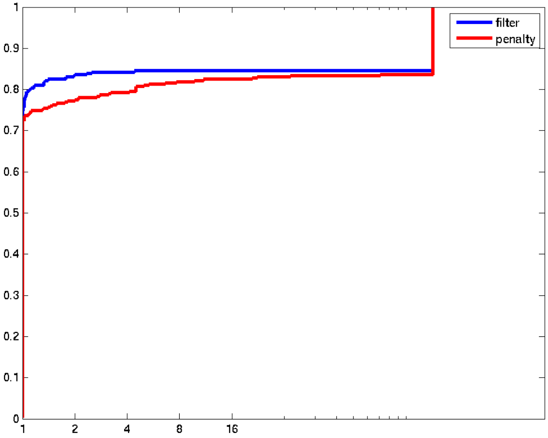
Filter $\mathcal{F}$: list of non-dominated pairs (h_l, f_l)

- new x_{k+1} acceptable to filter $\mathcal{F}$, iff
  1. $h_{k+1} \leq h_l \; \forall l \in \mathcal{F}$, or
  2. $f_{k+1} \leq f_l \; \forall l \in \mathcal{F}$
- remove redundant entries
- reject new x_{k+1}, if $h_{k+1} > h_l$ & $f_{k+1} > f_l$ & reduce trust region $\Delta = \Delta/2$



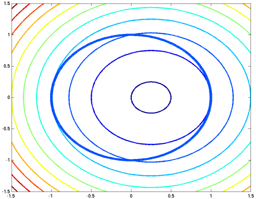
$\Rightarrow$ often accept new x_{k+1}, even if penalty function increases

Filter vs. Penalty



... quite similar, luckily filter still wins!

Maratos Effect in Filter Methods



Filter methods suffer Maratos Effect:

$$\begin{array}{ll} \text{minimize} & 2(x_1^2 + x_2^2 - 1) - x_1 \\ \text{subject to} & x_1^2 + x_2^2 - 1 = 0 \end{array}$$

SQP step near x_0 near $(1, 0)$ increases objective & constraints:
$f_1 > f_0$ and $h_1 > h_0$ Newton step rejected by filter
$\Rightarrow$ need second-order correction (SOC) steps

SOC steps are cumbersome ... can we avoid them?

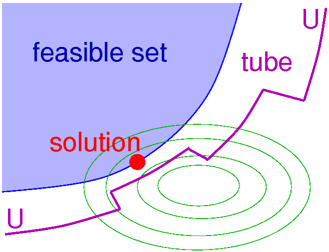
Idea: Use non-monotone filter ... generalizes standard filter.

Funnel for Optimization

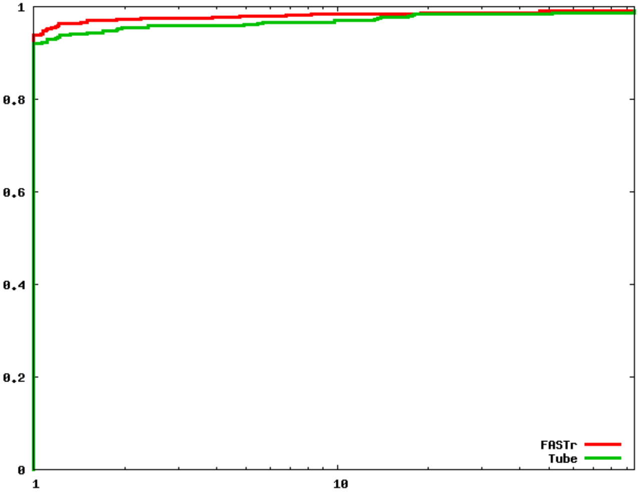
Predates filter methods (tube); recently see [Gould & Toint, 2007]

Idea: accept step that does not deteriorate max. infeasibility

- initialize tube as
  $U = \max\{1.25\|c(x_0)^-\|, 100\}$
- accept x_{k+1}, if $\|c(x_{k+1})^-\| < U$
- if no sufficient f-reduction, then
  $U = \max\left\{0.9U, U - 0.1\text{ared}\right\}$,
  where
  $\text{ared} = \max\left\{10^{-4}, h_k - h_{k+1}\right\}$



Filter vs. Tube



... very similar, but tube is easier!

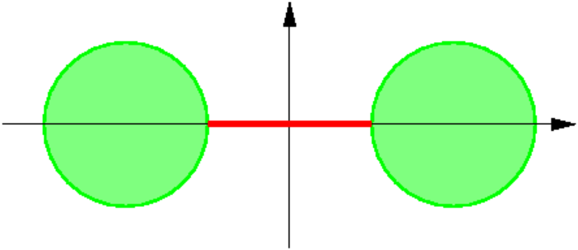
Sometimes Things Go Badly Wrong!

1. exception handling
   - floating point (IEEE) exceptions from function evaluations
   - unbounded problems: $f(x) \to -\infty$ for $c(x) \geq 0$
2. only get local solutions or stationary point global optimization hard for 100 vars
   - (locally) inconsistent problems
   - suboptimal, or only stationary

... sometimes ignored by optimization community ...

Locally Inconsistent Problems

NLP may have no feasible point
... need to detect this quickly, e.g. mixed-integer problems



feasible set: intersection of circles

- Any point on red line "proofs" local infeasibility
- Jacobian of two constraints is linearly dependent

Can we identify infeasibility quickly?

Locally Inconsistent Problems

Local QP/LP approximation inconsistent:

$$\left\{ s \;:\; c_k + A_k^T s \geq 0, \;\text{ and }\; \|s\| \leq \Delta_k \right\} = \emptyset$$

Formulate feasibility problem:

1. Divide constraints into infeasible $\mathcal{I}$ and rest $\mathcal{I}^c$

$$\mathcal{I} := \left\{ i \;:\; c_i + a_i^T s < 0 \right\}$$

2. Minimize infeasibility subject to remaining constraints

$$\begin{cases} \underset{x}{\text{minimize}} & \sum_{i \in \mathcal{I}} \left(-a_i^T s \right) - s^T \hat{H} s / 2 \\ \text{subject to} & c_i + a_i^T s \geq 0, \quad \forall i \in \mathcal{I}^c \end{cases}$$

   where $\hat{H} = \sum y_i \nabla^2 c_i$ Hessian of constraints.

3. Switch between feasibility restoration & optimization.

Observe fast (quadratic) local convergence.

Online Optimization Tools

- NEOS server: web-based solvers
  `http://www-neos.mcs.anl.gov/`
- NEOS guide & wiki: information
  `http://www-fp.mcs.anl.gov/OTC/Guide/`
  `http://wiki.mcs.anl.gov/NEOS/`
- TAO: toolkit for advanced optimization
  `http://www.mcs.anl.gov/tao/`
  Parallel optimization using PETSc
- COIN-OR project: open-source solvers
  `http://www.coin-or.org/`

Conclusions & Future Work

Optimization has more & cooler acronyms than Linear Algebra!

FASTr: Flexible Active-Set Trust-Region Framework

- Local methods (step computation)
  - sequential quadratic programming (SQP)
  - sequential linear/quadratic programming (SLQP)
  - sequential regularized LP/QP matrix free possible (RLQP)

  Solvers: `BQPD` (Fletcher), `Clp` (COIN-OR), `MA57` (Harwell)
- Forcing Strategy (step acceptance)
  - penalty function
  - filter methods as alternative to penalty functions
  - non-monotone filter ... avoid Maratos effect???
  - tolerance tube: easier than filter; almost as efficient
- future developments
  - more subproblem solvers: LP/QP/EQP (PARDISO, SCIP for LP)
  - heuristics for nonlinear optimization

Mixed-Integer Nonlinear Program (MINLP)

... but MINLP also stands for

Most Interesting Nincompoops Love Pabst
Mostly Irrelevant Nonsense and Ludicrously Pompous
Monumental Imbibing Now Lauded Posthumously
Masters of Impudent Nepotistic Lies and Prevarications
Multiple Injury Nobbles Lonely Person
Mission Impossible Needs Loads of Patronage
Mission Impossible Nearing Limitless Perfection

See ya at the Victoria!